



Elements Enabling High-level Communication in Power Systems

Kullmann, Daniel; Bindner, Henrik W.

Publication date:
2011

Document Version
Publisher's PDF, also known as Version of record

[Link back to DTU Orbit](#)

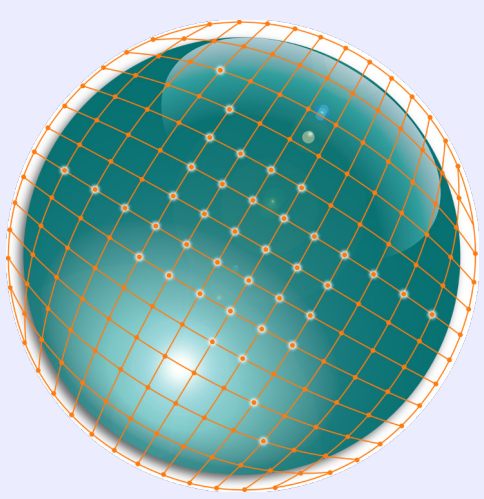
Citation (APA):
Kullmann, D., & Bindner, H. W. (2011). *Elements Enabling High-level Communication in Power Systems*. Poster session presented at 4th International Conference on Integration of Renewable and Distributed Energy Resources, Albuquerque, New Mexico, United States.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

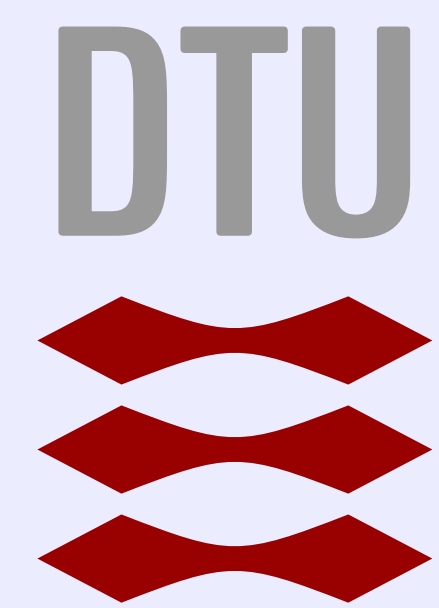
- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.



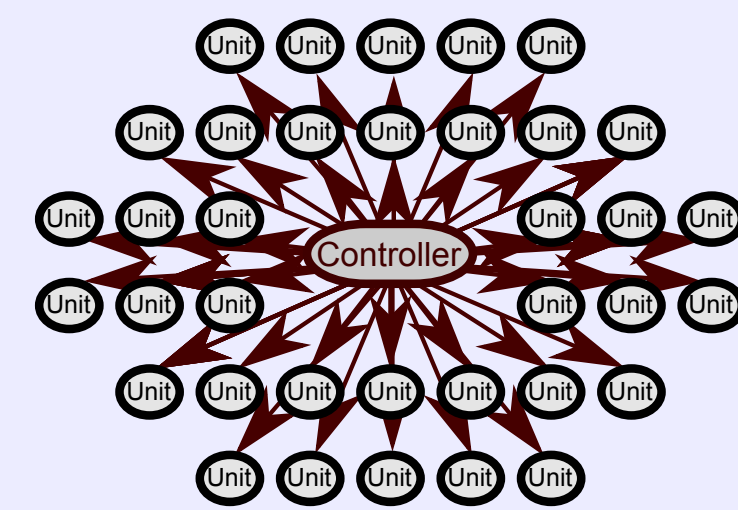
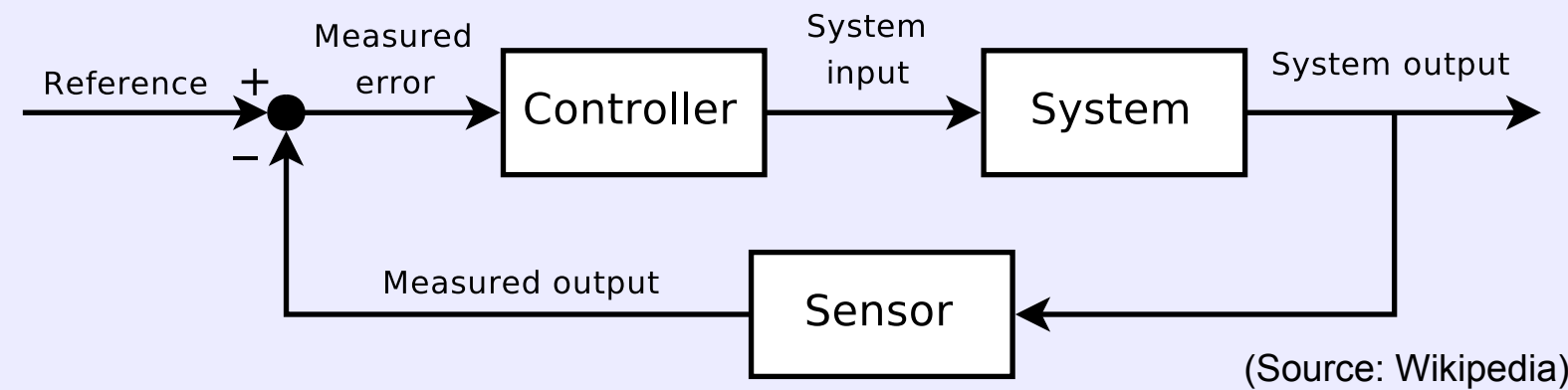
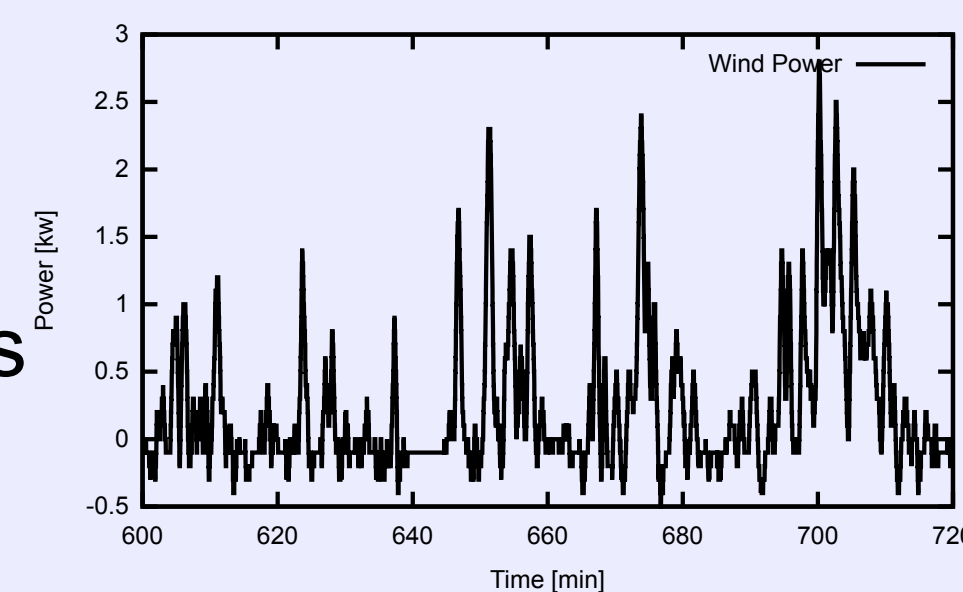
Elements Enabling High-level Communication in Power Systems

Daniel Kullmann, Henrik Bindner; Risø DTU, Denmark



Background

- fluctuating power of renewable resources
- managing large numbers of components
 - generation and consumption
- control implies communication
- conventional control: closed-control loops
- communication links are often unreliable
 - no bandwidth guarantees
 - no latency guarantees
 - fails sometimes completely



Communication

Requirements on communication :

- scalable
- flexible and extensible
- device-type-agnostic
- robust against failure, misuse and attacks
- cost-effective
- work with unreliable communication links

role-based approaches

IT security

use existing communication links

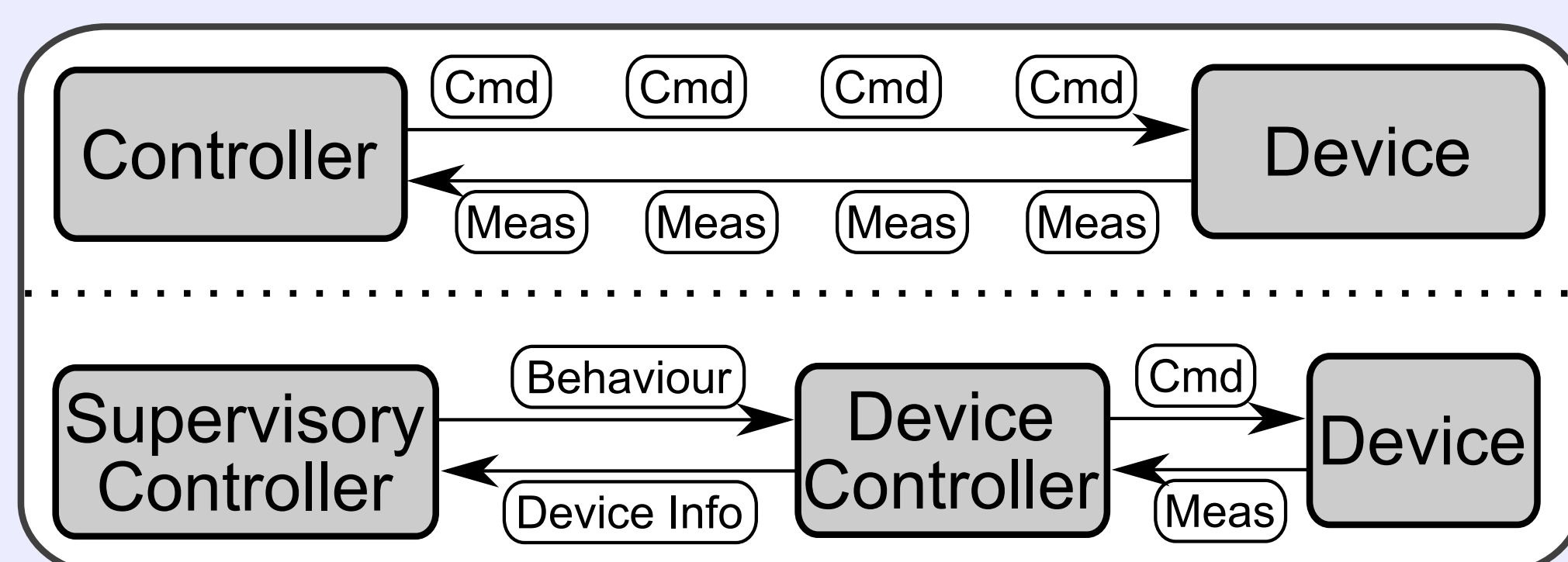
Split up control:

- *device control*: closed-control loop
- *supervisory control*: more abstract comm.

High-level communication
more abstract, more general, more expressive

High-level communication

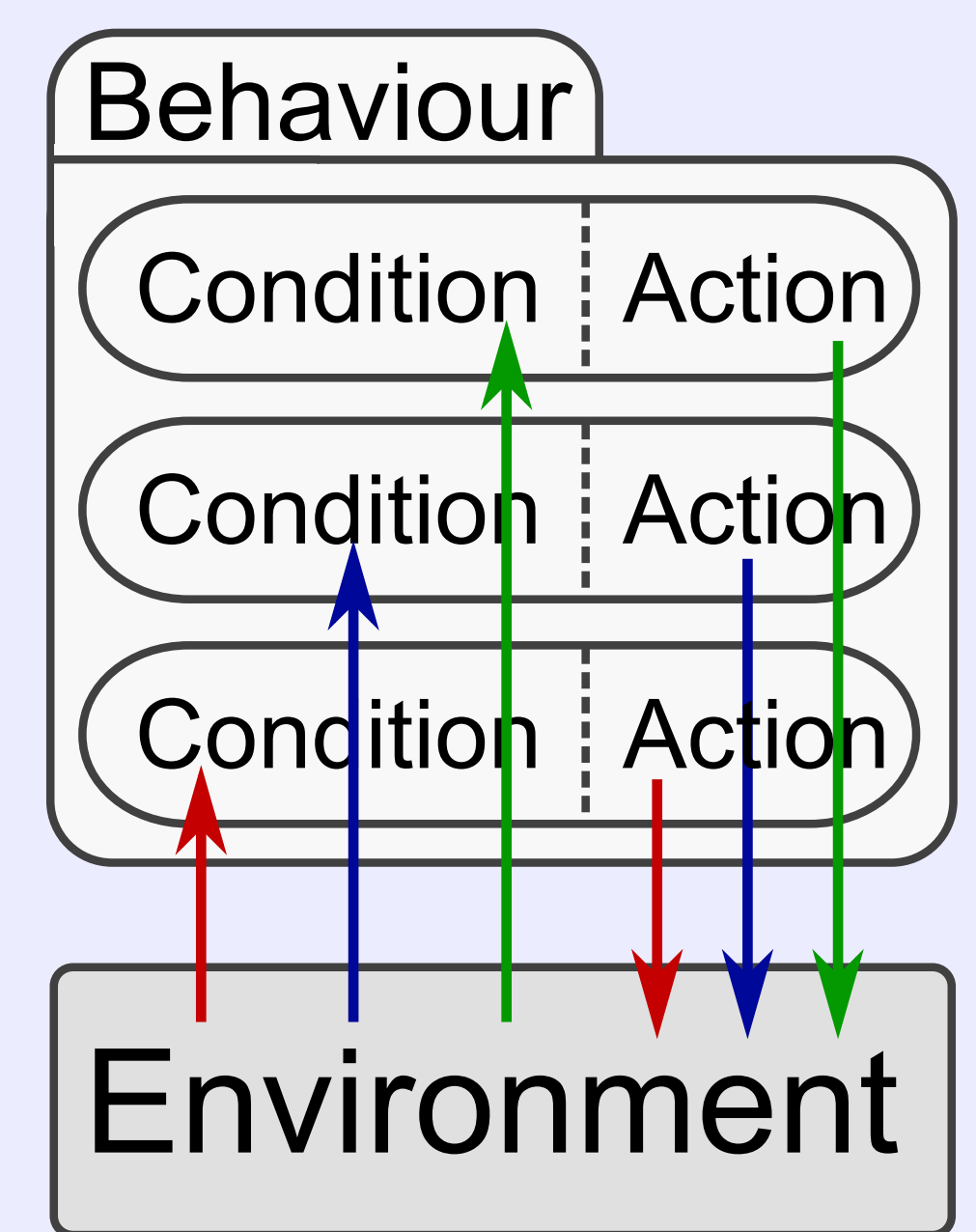
low-level	measurements/ setpoints	device- specific	values
high-level	services/ behaviours	device- independent	ontologies



Behaviour descriptions

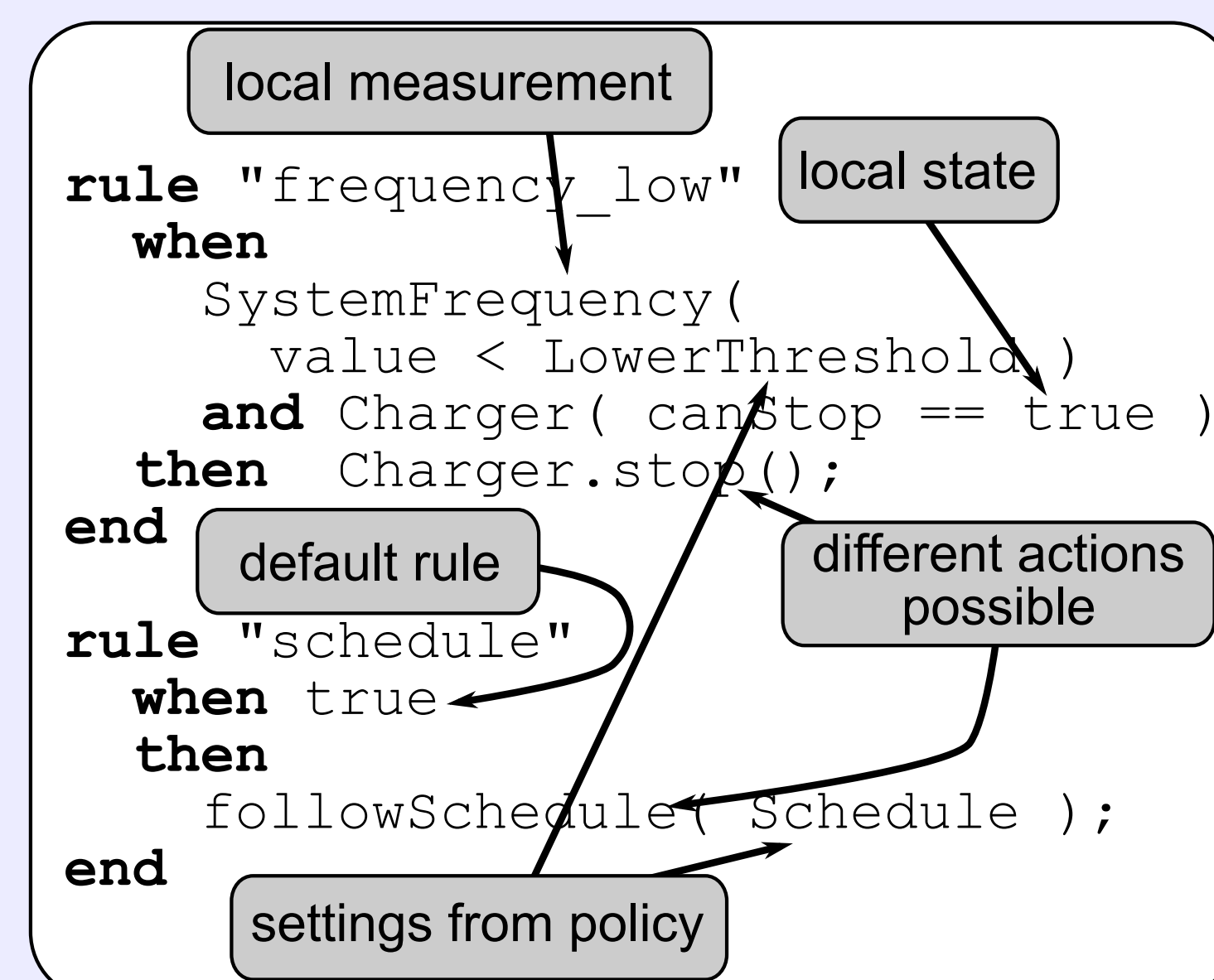
- tells component how to behave
- "Off-line" behaviour:
 - send behaviour first
 - then behaviour is acted upon
- component acts on locally observable events, e.g.
 - system frequency
 - power price (broadcast)
- component also acts locally

- implementation options:
 - very general: "keep system stable"
 - very specific: "set set-point to value x"
 - rule-based systems!



Behaviour Rules - Policies

- set of if-then rules
- conditions refer to local measurements
- many possible action types
 - set-points
 - process control
 - schedule
 - activation of other rules
 - etc.
- rule-system is flexible: can react to different situations
- rule system is extensible: many different action types



Standards

Standards are necessary to make unified access to components possible. There are two main standard families for Smart Grid communication.

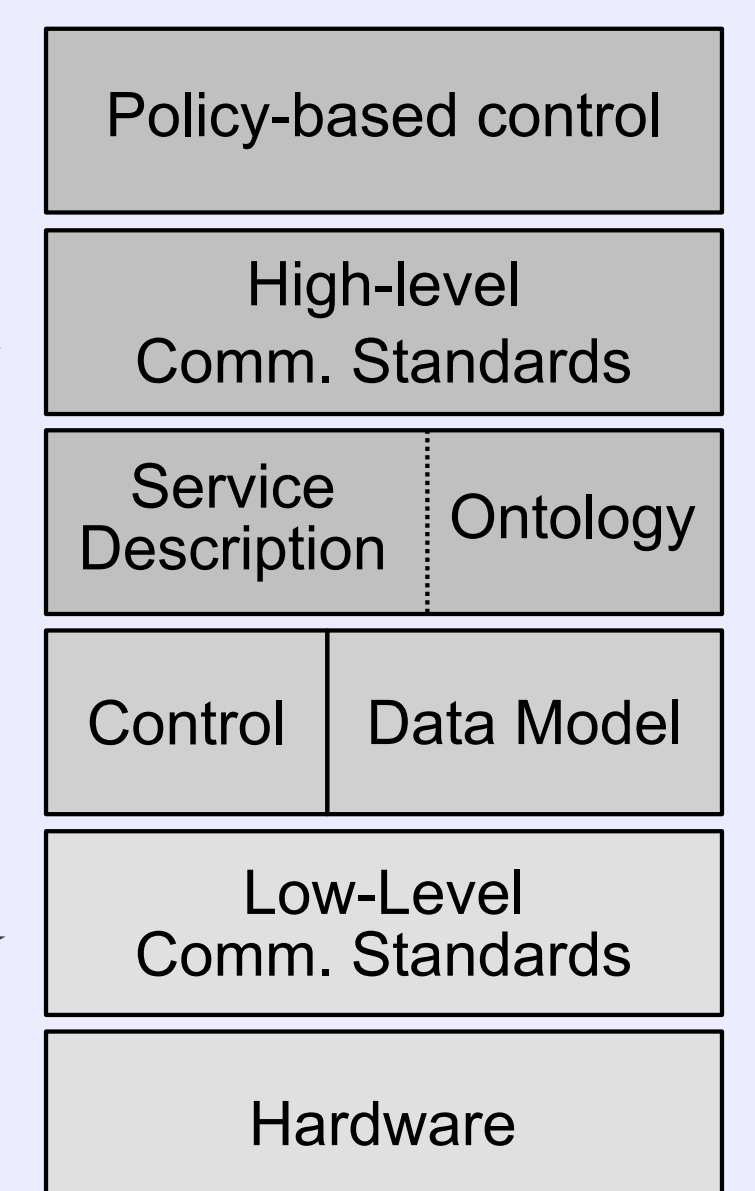
IEC 61970 / IEC 61968:

- also known as (CIM)
- communication about components
- provides a power system ontology
- used to define message formats

IEC 61850:

- relatively low-level
- communication with components

Policy communication stack



Example

- charging of electric vehicles
- 3 levels of hierarchy:
 - schedule
 - dynamic power price
 - frequency
- Settings object is sent along with policy
- different threshold values for price and frequency
- prevent synchronous responses of vehicles to changes
- rule set can be completely changed for each EV
- possible to account for individual constraints of the vehicle

```
rule "frequency_low"
when
  SystemFrequency( value < Settings.LowFrequency )
  and charger : Charger( canStop == true )
then
  charger.stop();
end
rule "frequency_high"
when
  SystemFrequency( value > Settings.HighFrequency )
  and charger : Charger( canStart == true )
then
  charger.start();
end
rule "price_low"
when
  PowerPrice( value < Settings.LowPrice ) and
  charger : Charger( canStart == true )
then
  charger.start();
end
rule "price_high"
when
  PowerPrice( value > Settings.HighPrice ) and
  charger : Charger( canStop == true )
then
  charger.stop();
end
rule "schedule"
when
  charger : Charger() # charger is present
then
  charger.followSchedule( Settings.Schedule );
end
```

Interaction Protocol

The way the communication between supervisory controller and supervised component is structured must also be standardised. This is done by defining a protocol the parties have to adhere to.

In principle, it is a three-step process:

1. the controlled unit sends information about itself to the supervisory ctrl.
2. The controller generates a policy for the unit and sends it to the unit, which has to accept it
3. The unit activates the policy

This process is restarted after some time.

